MEMORY MATCHING NETWORKS FOR GENOMIC SEQUENCE CLASSIFICATION

Jack Lanchantin, Ritambhara Singh, & Yanjun Qi
Department of Computer Science
University of Virginia
Charlottesville,VA 22903, USA
{jjl5sw,rs3zz,yq2h}@cs.virginia.edu

ABSTRACT

When analyzing the genome, researchers have discovered that proteins bind to DNA based on certain patterns on the DNA sequence known as "motifs". However, it is difficult to construct motifs for protein binding location prediction due to their complexity. Recently, external learned memory models have proven to be effective methods for reasoning over inputs and supporting sets. In this work, we present memory matching networks (MMN) for classifying DNA sequences as protein binding sites. Our model learns a memory bank of encoded motifs, which are dynamically learned, and then matches a new test sequence to each of the motifs to classify the sequence as a binding or non-binding site.

1 INTRODUCTION

In genomics, Transcription Factors (TFs) are proteins which bind to certain areas on a DNA sequence and in turn control gene regulation. Thus, predicting the Transcription Factor Binding sites (TFBSs), or locations where TFs bind on the genome, is particularly useful for understanding genomic processes and improving human health. Biologists have discovered that the binding of a TF is triggered by local sequential patterns within TFBSs, known as "motifs" (Stormo, 2000). As a result, researchers tried to predict TFBSs by computationally constructing motifs using position weight matrices (PWMs) which best represented the positive binding sites, and then compared the test sequence to the PWMs to see if there is a close match. However, it is difficult to find accurate PWMs due to the wide variety of TFBS sequences. Additionally, a TFBS may be influenced by a combination of different motifs. In this work, we attempt to create a memory bank of fully learnable motifs, which we can then read from to compare with and classify a new test sequence.

On the TFBS task, researchers initially used PWM-matching approaches (Stormo, 2000), which were computationally created motifs for comparison to a test sequence. This was later outperformed by a convolutional neural network (CNN) model which could learn PWM-like filters (Alipanahi et al., 2015). In our model, we use a learned memory bank of PWM-like matrices which we directly use for matching against the test sequence. We run our model on a baseline TFBS dataset and compare against PWM, CNN, and LSTM approaches.

Recently, deep learning models that use external memory have emerged as promising methods for many tasks (Graves et al., 2014; Sukhbaatar et al., 2015; Vinyals et al., 2015; Miller et al., 2016). These studies concentrate on reasoning using external memory based on the input to produce some target. In our work, we focus on creating a model which uses comparison over a learned memory bank of motifs for classification. Our work is closely related to the "matching network" (MN) model by Vinyals et al. (2016), where they train a differentiable nearest neighbor model to find the closest matching image from a support set on a new unseen image. We introduce Memory Matching Networks (MMN), where we modify the MN model by replacing the support set with a dynamic memory support set for the TFBS classification task. The key difference is that our MMN model is for a general classification setting (i.e. not one-shot) and we seek to instead *learn* the support set (memory templates), which remains constant for every new test classification. We believe that the dynamically learned memory can function similarly to the motifs that traditional PWM methods used. Our preliminary models show that the learned memory can find the binding patterns better than traditional

deep learning methods, and also provide similar explanatory patterns that are predominantly used in bioinformatics.

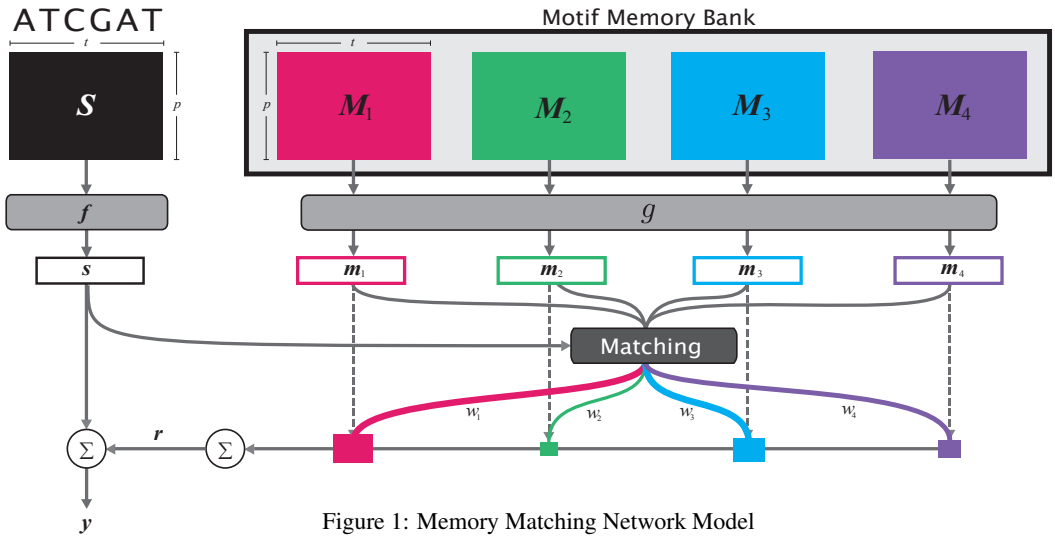

Figure 1: Memory Matching Network Model

2 Memory Matching Network Model for Classification

Given a DNA sequence S (composed of characters A,C,G,T) of length t and a Transcription Factor protein of interest, we want to classify S as a positive or negative binding site for this particular TF (i.e. binary classification task). To do this, we seek to match S to a bank of ℓ learned memory matrices, where each matrix is loosely representative of a motif. Our output classifier is then based on a linear combination of the motifs and their matching scores.

Each memory matrix M_i is learned via a lookup table $\in \mathbb{R}^{p \times t}$ with a constant input integer at each position (e.g. 1 as input to the first position and t as input to position t). To learn each of the ℓ memory templates, we use a separate lookup table to produce ℓ memory matrices $\{M_1, ..., M_\ell\} \in \mathbb{R}^{p \times t}$. Similarly, we encode the input sequence using another lookup table $\in \mathbb{R}^{p \times 4}$ with the 4 possible nucleotide inputs to produce input matrix $S \in \mathbb{R}^{p \times t}$. We map the encoded input $S \in \mathbb{R}^{p \times t}$ and memory matrices $\{M_1, ..., M_\ell\}$ into vectors using two separate functions, f and g, respectively:

$$\begin{aligned} \boldsymbol{s} &= f(S) \\ \boldsymbol{m}_i &= g(g'(M_i)) \text{ for } i \text{ in } 1, ..., \ell \end{aligned} \tag{1}$$

f is a bidirectional attention LSTM on S, where we compute a context vector $\boldsymbol{s} \in \mathbb{R}^d$ based on the attention weights α_i as in Bahdanau et al. (2014). g' is a bidirectional attention LSTM on each M_i, and shares the weights with f. g is another bidirectional LSTM (no attention) that takes the outputs of g' as input to encode dependencies among the motifs, producing final memory vectors $\{\boldsymbol{m}_1,...,\boldsymbol{m}_\ell\} \in \mathbb{R}^d$.

Once we have the embedding vectors, we want to compare the original input sequence to the memory templates for classification. We hypothesize that different memory templates contribute differently. The importance, or weight w_i, of each memory matrix M_i is calculated using a normalized cosine similarity score between $\boldsymbol{s}$ and each $\{\boldsymbol{m}_1,...,\boldsymbol{m}_\ell\}$, which is the matching part of the network:

$$w_i = \frac{\exp\Big(K[\boldsymbol{s}, \boldsymbol{m}_i]\Big)}{\sum_j \exp\Big(K[\boldsymbol{s}, \boldsymbol{m}_j]\Big)} \tag{2}$$

where K is cosine similarity. Then a read vector r, is computed as a linear combination of the memory embeddings multiplied by their respective matching weights. This allows for the sequence S to not be forced to match to a particular motif, but rather a combination of them:

$$\boldsymbol{r} = \sum_i w_i \mathbf{m}_i \tag{3}$$

The read vector r and original input embedding vector s are then linearly transformed, added together, and then fed through a softmax equation to get $\mathbf{y}$, the probability of S being a positive TFBS:

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathrm{W}_s \boldsymbol{s} + \mathrm{W}_r \boldsymbol{r} \\ \mathbf{y} &= softmax\big(\tilde{\mathbf{y}}\big) \end{aligned} \tag{4}$$

Table 1: TFBS Binary Classification Results

Model	Mean AUC	Median AUC	Stdev
PWM	0.850	0.876	0.120
LSTM	0.910	0.932	0.056
CNN	0.918	0.940	0.084
MMN (ours)	0.929	0.947	0.067

3 EXPERIMENTS

We ran our MMN model on the 61 leukemia cell TF datasets which had a training set of at least 10,000 sequences from Alipanahi et al. (2015). Each TF dataset has exactly 1,000 testing sequences. All training and testing sets have an even positive/negative TFBS sequence split. Each sequence sample S is 101 length ($t = 101$) and composed of DNA-base characters (A,C,G,T). Since there is a separate dataset for each different TF, we train a separate model for each TF. In other words, each model constructs a memory bank for that particular TF. We then aggregate the accuracy results over all TFs for model comparison. In our experiments, we tuned the following hyperparameters: number of memory matrices $\ell \in \{2, 4, \mathbf{8}, 16\}$, memory matrix column size $p \in \{2, \mathbf{4}, 8, 16\}$, memory embedding vector size $d \in \{32, 64, \mathbf{128}\}$, with the best parameters in bold.

3.1 ACCURACY RESULTS

We compare our method to the baseline PWM-motif matching approach from Machanick & Bailey (2011), as well as the CNN model in Alipanahi et al. (2015), and a regular LSTM model in Lanchantin et al. (2016). The results are shown in Table 1. Our model significantly outperforms the baseline of constructed PWMs (based on a pairwise t-test). Our model performs slightly better than the other two baseline deep learning models, including the CNN, which should be able to extract motifs automatically via filters. We believe that this is an important result to show that matching network models are powerful for classification.

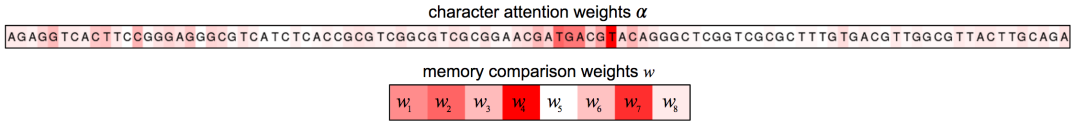

Figure 2: Visualizing the Model: Character and memory weights for a positive ATF1 binding sequence

3.2 VISUALIZING AND INTERPRETING THE MODEL

In medical classification tasks, if a model is accurate, yet hard to interpret, biomedical researchers may be reluctant to use it. Thus, we seek to understand our model via visualization. While it's difficult to directly correlate the memory matrices with DNA characters since they are in an embedded space, we can look at individual test sequences and see what the model focuses on. We do this by viewing the attention weights $\{\alpha_1, ..., \alpha_t\}$ from f to see which characters are most important, and also the memory matching weights $\{w_1, ..., w_\ell\}$ to see which memory matrices closely match this sequence. Figure 2 shows an example of our visualizations where the sequence is a positive TFBS sequence from the "ATF1" TF. We can see that the character-level α attention mostly concentrates on the subsequence *TGACGTA* in the middle, which is a "known" motif for ATF1 (Mathelier et al., 2016). Seeing which particular memory templates match the test sequence via w_i doesn't tell us anything, but it may be of use for comparing multiple sequences to see if they match certain memories. We plan to explore the interpretability of the learned memory in future work.

4 CONCLUSION

In this work, we introduced memory matching networks (MMNs) for learning a set of memory matrices which are matched to a new test sequence for classification. We applied our model on the binary classification task of predicting the binding sites of Transcription Factor proteins, where a memory of learned "motifs" is beneficial for classifying sequences. We showed that it outperforms the baseline models of PWM-matching and a CNN, as well as provided a way to visualize the predictions. We hope that this work will provoke more research on memory-based models for genomic sequence tasks, where it is a fitting setting. Although we use this on a medical task, we think that the memory matching network framework may be of use in any application where classification can be done by matching to memory, such as on text or images.

REFERENCES

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Jack Lanchantin, Ritambhara Singh, Beilun Wang, and Yanjun Qi. Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks. *arXiv preprint arXiv:1608.03644*, 2016.

Philip Machanick and Timothy L Bailey. Meme-chip: motif analysis of large dna datasets. *Bioinformatics*, 27(12):1696–1697, 2011.

Anthony Mathelier, Oriol Fornes, David J Arenillas, Chih-yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, et al. Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 44(D1):D110–D115, 2016.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

Gary D Stormo. Dna binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pp. 2440–2448, 2015.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.